Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 46 (2015) 1510 – 1517
Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

A Proactive Approach to Reversible Data Hiding in Encrypted Images

Nirmal S. Nair[a,]*, Tojo Mathew[a], Neethu A.S.[a], Viswajith P. Viswanath[a], Madhu S. Nair[a], M. Wilscy[a]

[a]*Department of Computer Science, University of Kerala, Kariavattom, Thiruvananthapuram-695581, Kerala, India*

Abstract

We present a reversible data hiding method for encrypted images that guarantees reversibility, i.e., the exact recovery of secret data and cover image by the receiver. This work is based on Zhang's reversible data hiding in encrypted images. Spatial correlation of pixels in the cover image is exploited for data hiding. In the proposed method, the sender proactively chooses suitable blocks in the cover image to hide secret data. The stego image is then encrypted so that privacy of cover image as well as secret data is protected. Zhang's method attains reversibility only with sufficiently large block size, thereby sacrificing payload capacity. The proposed method achieves reversibility at all block sizes, while significantly improving the embedding capacity over many of the state-of-the-art methods.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

*Keywords:*Image encryption; image recovery; pixel classification; reversible data hiding; spatial correlation;

1. Introduction

Data hiding is the technique of hiding secret data into a cover media. This might often produce a distorted version of the cover media, known as stego media. Several data hiding methods, using digital images as cover media, have been proposed in the past[1,2,3]. Based on the extent to which the original cover image can be recovered from the stego

* Corresponding author. Tel.: +91-9895403183.
*E-mail address:*nirmal.nair@outlook.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.071

image, data hiding techniques can be classified into: irreversible (lossy) data hiding, and reversible (lossless) data hiding. Reversible data hiding (RDH)[5,13,14,17] allows the cover image to be completely reconstructed from the stego image, along with extraction of embedded secret data, by the receiver. Most commonly used approaches for RDH are histogram shifting, difference expansion and compression[11]. For example, Ni et al.[5] and Chen et al.[15] use the histograms obtained from cover image for embedding data by histogram shifting. Difference expansion methods[6,16] expand the variation in adjacent pixel values to accommodate additional data. Methods [4,7] create room for payload by lossless compression of suitable regions of the image. RDH techniques find applications in remote sensing, medical imaging, forensics etc. where accuracy of the image content is highly sought.

Recently, reversible data hiding in encrypted images[8,9,10] (RDH-EI) has attracted much attention in application areas where privacy of the cover image is to be protected. Cloud storage of images, medical imaging, military and law enforcement etc. are some of the application areas where privacy of the cover images is equally important as the data hidden in them. For example, nowadays there are many web service providers offering cloud based storage space for personal data. Images, videos etc. stored in this cloud space are encrypted. Additional information such as source details, content description, tags etc. can be embedded into the encrypted content, which will enable the web service provider to manage those files without infringing the content privacy. Medical imaging is another field where images are encrypted to ensure patient privacy. Using RDH-EI, confidential information like patient details, diagnosis etc. can be hidden within the image. Error-free retrieval of the hidden data and restoration of the original image are crucial for such applications.

Ma et al.[12] defined two possible approaches for RDH-EI. First approach is to find room for additional data after encrypting the image, referred as 'Vacating Room After Encryption' (VRAE). Methods [8-10] follow this approach. The second approach is to reversibly reserve required amount of space for additional data before encrypting the image. After encryption, additional data is hidden into the space reserved. This is referred as 'Reserving Room Before Encryption' (RRBE). In the proposed method, we adopt the VRAE approach and address limitations of other VRAE methods [8-10]. The major limitation of these methods is conditional reversibility, i.e., recovery of the cover image and error-free extraction of hidden data are not guaranteed in all scenarios. In these methods, true reversibility and payload capacity are a trade-off to choose between. To ensure true reversibility, payload capacity has to be compromised. Increasing the payload capacity beyond a limit will cause errors during data extraction. The proposed method provides significant improvement in payload capacity while ensuring true reversibility in all scenarios.

2. Related work

One of the pioneering techniques in RDH-EI has been proposed by Zhang[8]. Later, Hong et al.[9] improved this method by enhancing the accuracy of the extraction process. In [8] and [9], an 8-bit cover image I of size $m \times n$ is divided into k non-overlapping blocks of size $s \times s$. The pixels in each block B^i is randomly classified into two sets S_0 and S_1, using a data hiding key K_D. Each block B^i can carry at most one bit. A random sequence r of size $m \times n \times 8$ bits is generated using an encryption key K_E. I is encrypted by performing a bitwise XOR operation with r.

$$I' = I \oplus r \tag{1}$$

Secret data is embedded bit-by-bit in row-major order of the blocks. If the bit to embed in a block is 0, the 3 LSBs of pixels in S_0 are flipped. If the bit to embed is 1, the 3 LSBs of pixels in S_1 are flipped. Every block is either embedded with 0 or 1. Let I'' denote the encrypted image embedded with secret data.

The receiver requires K_E, K_D, and s for data extraction and image recovery. The receiver decrypts the encrypted image I'' using K_E and partitions it into k non-overlapping blocks of size $s \times s$. The pixels in each block are classified into two sets S_0 and S_1, using K_D. The receiver processes the blocks in the same order as the sender. The objective of the receiver is to determine, for each block, whether the pixels in S_0 or the pixels in S_1 were flipped by the sender. Let H^i represent the blocks of the decrypted stego image. $\widetilde{H}^i$ is obtained by flipping pixels in S_0 and $\widehat{H}^i$ is obtained by flipping pixels in S_1. If the embedded bit in H^i is 0, $\widetilde{H}^i$ matches the original block B^i and $\widehat{H}^i$ matches the fully flipped block $\bar{B}^i$. If the embedded bit in H^i is 1, $\widetilde{H}^i$ matches $\bar{B}^i$ and $\widehat{H}^i$ matches B^i.

$$\bar{B}^i_{u,v} = B^i_{u,v} \oplus 11111111_2, (u,v) \in S_0 \text{ or } S_1 \tag{2}$$

The pixels in the original block B^i have better spatial correlation than $\bar{B}^i$. Zhang proposed a fluctuation function to capture the spatial correlation of pixels in each block.

$$f = \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|p_{u,v} - \frac{p_{u-1,v} + p_{u+1,v} + p_{u,v-1} + p_{u,v+1}}{4}\right| \tag{3}$$

where $p_{u,v}$ is the pixel value at location *(u,v)* in the block. Hong et al. improved the fluctuation function by including the border pixels of blocks as well as taking into consideration the correlation between adjacent blocks.

$$f = \sum_{u=1}^{s}\sum_{v=1}^{s-1}|p_{u,v} - p_{u,v+1}| + \sum_{u=1}^{s-1}\sum_{v=1}^{s}|p_{u,v} - p_{u+1,v}| \tag{4}$$

By applying the fluctuation function, the fluctuation of $\tilde{H}^i$ and $\hat{H}^i$ is calculated as $\tilde{f}^i$ and $\hat{f}^i$, respectively. Let the difference between $\tilde{f}^i$ and $\hat{f}^i$ be denoted as A^i.

$$A^i = \tilde{f}^i - \hat{f}^i \tag{5}$$

The block having lower fluctuation value will be regarded as the original block. If $A^i < 0$, a bit 0 is extracted and $\tilde{H}^i$ is regarded as the original block. Otherwise, a bit 1 is extracted and $\hat{H}^i$ is regarded as the original block. In [9], $|A^i|$ is calculated for all blocks and sorted in descending order. A higher value for $|A^i|$ implies that the block B^i is more dissimilar to $\bar{B}^i$. [9] uses this sorted order of $|A^i|$ for data extraction and image recovery of blocks. It also uses a 'side-match' mechanism for improving the accuracy.

3. Proposed method

Methods [8] and [9] assume that the pixels of an original block B^i will have better spatial correlation than the corresponding flipped version $\bar{B}^i$. The chosen block size *s* should be sufficiently large for this assumption to hold true for all blocks. Using a large block size leads to reduction in embedding capacity, whereas a smaller block size results in more blocks that violate this assumption. Such blocks, for which the flipped version has better correlation than the original one, cause errors during data extraction at the receiver. These errors can be eliminated if the sender identifies and blacklists such blocks instead of blindly embedding secret data into all available blocks. The proposed method expands on this idea, thus guaranteeing reversibility for any block size.

3.1. P-Z-N classification of blocks

Let f^i and $\bar{f}^i$ represent the fluctuation values of blocks B^i and $\bar{B}^i$, respectively. The difference D^i is obtained as:

$$D^i = \bar{f}^i - f^i \tag{6}$$

Based on the value of D^i, block B^i can be classified into one of three classes: Positive *(P)*, Zero *(Z)*, or Negative *(N)*.

$$B^i \in \begin{cases} P, & D^i > 0 \\ Z, & D^i = 0 \\ N, & D^i < 0 \end{cases} \tag{7}$$

P-blocks are ideal for data embedding, since the embedded data bit can be correctly extracted by the receiver. *Z-blocks* do not have data carrying capacity, and hence are not used for data hiding. Any data bit embedded in *N-blocks* will be incorrectly extracted by the receiver. This also leads to an incorrect image reconstruction. Therefore, reversibility of cover image and secret data can be achieved if only *P-blocks* are used for data hiding.

(a) (b) (c) (d) (e)

Fig 1. Test images: (a) Lena, (b) F-16, (c) Man, (d) Baboon, and (e) Spine.

3.2. Improved fluctuation function

An ideal fluctuation function should classify all blocks as P, regardless of the block size used. As the block size decreases, the total number of pixels available to evaluate the fluctuation decreases, hence, the accuracy of the fluctuation function decreases. Based on empirical results, we propose a new fluctuation function which gives a better estimation than equations (3) and (4).

$$f = \sum_{u=1}^{s}\sum_{v=2}^{s-1} 2 \times p_{u,v} - (p_{u,v-1} + p_{u,v+1}) + \sum_{u=2}^{s-1}\sum_{v=1}^{s} 2 \times p_{u,v} - (p_{u-1,v} + p_{u+1,v}) \tag{8}$$

Table 1 shows the experimental results obtained by applying the fluctuation functions (3), (4), and (8) on various images. $n(P)$, $n(Z)$, and $n(N)$ denote the number of blocks belonging to class P, Z, and N, respectively. The objective of the fluctuation function is to maximize $n(P)$. Equation (8) provides better results on almost all test images.

Table 1. Comparison of number of *P-blocks*, *n(P)* , obtained using fluctuation functions f_{Zhang} (3), f_{Hong} (4), and $f_{Proposed}$ (8).

Cover Image *C*	Block size $s \times s$	Total blocks k	*n(P)*		
			f_{Zhang}	f_{Hong}	$f_{Proposed}$
Lena (512 × 512)	12 × 12	1,764	1,761	1,761	**1,763**
	8 × 8	4,096	4,022	4,039	**4,052**
	4 × 4	16,384	13,435	14,574	**14,835**
F-16 (512 × 512)	12 × 12	1,764	1,693	**1,697**	**1,697**
	8 × 8	4,096	3,751	3,803	**3,812**
	4 × 4	16,384	11,542	13,051	**13,117**
Man (512 × 512)	12 × 12	1,764	1,729	1,722	**1,742**
	8 × 8	4,096	3,837	3,861	**3,945**
	4 × 4	16,384	12,127	13,683	**13,978**
Baboon (512 × 512)	12 × 12	1,764	1,603	1,668	**1,680**
	8 × 8	4,096	3,378	**3,633**	3,624
	4 × 4	16,384	10,740	**12,436**	12,191
Spine (490 × 367)	12 × 12	1,200	580	589	**594**
	8 × 8	2,745	984	1,024	**1,035**
	4 × 4	11,102	2,025	2,333	**2,377**

3.3. Reducing number of Z-blocks

For most images, $n(P)$ is much larger than $n(Z)$ and $n(N)$. However, in certain images like X-ray images, MRI images etc., $n(Z)$ will be significantly high. The test image Spine shown in Fig. 1(e) is an example. In such images, the change in intensity values between adjacent pixels is very small or often nil. Hence, there will be a large number of blocks, where all the pixels contained in it, has the same intensity value. For such blocks, fluctuation values of B^i and $\bar{B}^i$ evaluate to 0. Consequently, D^i becomes 0 for every such block, leading to a high value for $n(Z)$. Since *Z-blocks* are not used for data embedding, this gravely affects the image's data carrying capacity.

To reduce $n(Z)$, instead of randomly dividing the s^2 pixels of each block into S_0 and S_1, we randomly divide them into three sets S_0, S_1, and S_U, using key K_D. The pixels belonging to S_U are kept unchanged during data embedding. From (2), it is evident that introducing S_U causes fluctuation of $\bar{B}^i$ to become a non-zero value, thereby reducing $n(Z)$. Table 2 shows the difference in *P-Z-N* classification when S_0, S_1, and S_U is used.

Table 2. Effect of 3-set classification on *n(P)*, *n(Z)* and *n(N)* compared to 2-set classification of [8] and [9].

Cover Image *C*	Block size $s \times s$	Total blocks k	Sets	$f_{Proposed}$		
				n(P)	*n(Z)*	*n(N)*
Lena (512 × 512)	4 × 4	16,384	S_0, S_1	14,835	1,019	530
			S_0, S_1, S_U	**15,673**	132	579
F-16 (512 × 512)	4 × 4	16,384	S_0, S_1	13,117	2,639	628
			S_0, S_1, S_U	**15,115**	284	985
Man (512 × 512)	4 × 4	16,384	S_0, S_1	13,978	1,103	1,303
			S_0, S_1, S_U	**14,760**	180	1,444
Baboon (512 × 512)	4 × 4	16,384	S_0, S_1	**12,191**	231	3,962
			S_0, S_1, S_U	12,162	187	4,035
Spine (490 × 367)	4 × 4	11,102	S_0, S_1	2,377	8,687	38
			S_0, S_1, S_U	**10,943**	89	70

3.4. Sender side

Similar to [8] and [9], the cover image C is divided into k non-overlapping blocks of size $s \times s$. The pixels in each block are classified into three sets S_0, S_1, and S_U, based on a data hiding key K_D. *P-Z-N* classification of the blocks is performed using (6), (7), and (8). From (6), it can be seen that the receiver does not have the ability to distinguish between a *P-block* and *N-block*, since only $|D^i|$ is available to the receiver. To overcome this limitation, a threshold mechanism is used. Let L denote the values of D sorted in descending order. The absolute value of the smallest element in the list, $|L^k|$, is taken as the threshold t. Based on t, all blocks can be further classified into Viable *(V)*, Unviable *(U)*, or Exception *(E)* as follows:

$$B^i \in \begin{cases} V, & |D^i| \geq t \text{ and } B^i \in P \\ U, & |D^i| < t \\ E, & |D^i| \geq t \text{ and } B^i \in N \end{cases} \tag{9}$$

Block number corresponding to L^k should be communicated to the receiver. This is done by embedding extra data into the cover image, called 'header-data', in addition to the secret data. To increase capacity, instead of using $|L^k|$ as the threshold, we can use a block further up in the sorted list to get the threshold. This may increase $n(V)$ and $n(E)$, where $n(V)$ and $n(E)$ represent the number of *V-blocks* and *E-blocks*, respectively. However, block numbers of all *E-blocks*, should be included in header-data. Each block number is represented using w bits, where $w = \lceil \log_2 k \rceil$. The first w bits of header-data represent $n(E)$. The next *n(E)* × *w* bits are used to represent block-numbers of *E-blocks* in ascending order of D^i. Secret data carrying capacity of the image is then obtained as:

$$Payload\ Capacity = n(V) - (n(E) + 1) \times w \tag{10}$$

The sender then chooses a threshold t, which maximizes this capacity. Once the final *V-blocks* sequence is known the cover image is encrypted using (1) to obtain C'. Then, secret data is embedded into the identified *V-blocks* in the following order. First, header-data is embedded bit-by-bit in blocks of descending order of D^i. Then, secret data is embedded bit-by-bit in *V-blocks* in ascending order of block number, while skipping those previously embedded with header-data. *U-blocks* and *E-blocks* are embedded with bit 0 so that the original image block B^i can be recovered at the receiver by computing $\widetilde{H}^i$ of those blocks. The final encrypted stego image is referred as C''.

3.5. Receiver side

The receiver first decrypts the received image C'' using K_E to obtain C'. It is then partitioned into k non-overlapping blocks of size $s \times s$. The pixels in each block are then classified into three sets S_0, S_1, and S_U using K_D. Similar to [8] and [9], the receiver computes $\widetilde{H}^i$ and $\widehat{H}^i$ for each H^i. Their corresponding fluctuation is estimated as $\tilde{f}^i$ and $\hat{f}^i$, using equation (8). The difference between $\tilde{f}^i$ and $\hat{f}^i$ corresponds to $|D^i|$. The receiver sorts $|D^i|$ in descending order and uses this sorted order for extraction of header-data. The first w blocks are processed initially, which gives $n(E)$. Then, the next *n(E)* × *w* blocks are processed, which gives the block-numbers of blocks belonging to E. $|D^i|$ of the block corresponding to last *w* bits of header-data is taken as the threshold t.

After extraction of header-data, the sorted list is discarded and processing of blocks is done in row-major order. Using t and E, the receiver can be sure that for any block, if $|D^i| \geq t$ and does not belong to E, then it belongs to V. Also, if $|D^i| < t$, the block belongs to U. For extraction of secret data, the receiver processes only *V-blocks*, excluding those containing header-data. Similar to [8], A^i is calculated using equation (5). If $A^i < 0$, a bit 0 is extracted and $\widetilde{H}^i$ is regarded as the original block. Otherwise, a bit 1 is extracted and $\widehat{H}^i$ is regarded as the original block. Although, *U-blocks* and *E-blocks* do not contain secret data, they are required for exact recovery of cover image. Since *U-blocks* and *E-blocks* are embedded with bit 0, the original block B^i can be obtained by computing $\widetilde{H}^i$.

4. Experimental results

The proposed method has been tested with standard test images from SIPI database[18], including Lena, F-16, Man, Baboon, and Spine, shown in Fig. 1. The results are compared with state-of-the-art methods [8-10] that follow VRAE approach. Table 1 compares the performance of the proposed fluctuation function (8) to that of methods [8] and [9]. From the table, it is apparent that equation (8) gives better results than equations (3) and (4) for all images except Baboon.

The values for $n(P)$, $n(Z)$, and $n(N)$ obtained using 2-set and 3-set pixel classification are given in Table 2. In highly textured images like Baboon, $n(Z)$ is low irrespective of whether 2-set or 3-set classification is used. In smooth images like Spine, due to the constant background, the value for $n(Z)$ with 2-set classification will be very high. 3-set classification of the proposed method reduces $n(Z)$ and increases $n(P)$, thereby improving payload capacity. From the table, it can be seen that 3-set classification reduces $n(Z)$ for all test images.

The maximum payload capacities of methods [8-10], in case of true reversibility, are compared in Table 3. The capacity achieved by the proposed method exceeds the other methods for most of the images. Fig. 2 highlights the blocks of incorrect data extraction for images Lena and Man, when the block size $s = 6$. For Lena image, [8] produces 254 bit errors and [9] produces 125 bit errors. For Man image, [8] produces 624 bit errors and [9] produces 141 bit errors, whereas the proposed method gives no error at all. For most images, [8] requires a minimum block

size of 20 × 20 and [9] requires a minimum block size of 12 × 12 for data hiding to be reversible, whereas the proposed method imposes no such constraints.

(a) (b) (c) (d)

Figure 2. (a) Original image. Blocks of incorrect data extraction using (b) Zhang [8], (c) Hong et al. [9], and (d) Proposed method.

Table 3. Maximum embedding capacity (bits) without compromising reversibility

Cover Image *C*	Zhang [8]	Hong et al. [9]	Zhang [10]	Proposed method
Lena (512 × 512)	1,296	2,601	9,175	**11,767**
F-16 (512 × 512)	625	1,764	8,126	**9,933**
Man (512 × 512)	441	1,024	6,553	**7,451**
Baboon (512 × 512)	361	625	**2,097**	1,954
Spine (490 × 367)	289	529	–	**17,813**

5. Conclusion

The proposed method is well suited for medical, military, and forensic applications since error-free data extraction and image recovery is guaranteed, regardless of block size. The sender takes a proactive approach by identifying and avoiding all blocks that are incapable of carrying data. Payload capacity is significantly improved by

using smaller block sizes. The 3-set pixel classification technique further improves payload capacity for images having constant background, like medical images. Using the proposed method, it is possible to embed about 5-20 kb of secret data in a 512 × 512 grayscale image.

References

1. Van R. Anderson and F. Petitcolas, On the limits of steganography. IEEE J. of Selected Areas in Commun., 1998. 16(4), pp. 474-481.
2. R.G. van Schyndel, A.Z. Tirkel, and C.F. Osborne, A digital watermark. IEEE Int. Conf. Image Process., 1994. 2, pp. 86-90.
3. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, Techniques for data hiding. IBM Syst. J., 1996. 35(3-4), pp. 313-336.
4. M. Arsalan, S.A. Malik, A. Khan, Intelligent reversible watermarking in integer wavelet domain for medical images, J. Syst. Softw., 2012. 85(4), pp. 883-894.
5. Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, Reversible data hiding. IEEE Trans. Circuits Syst. Video Techn., 2006. 16(3), pp. 354-362.
6. J. Tian, Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Techn., 2003. 13(8), pp. 890-896.
7. M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, Lossless generalized-LSB data embedding. IEEE Trans. Image Process., 2005. 14(2), pp. 253-265.
8. X. Zhang, Reversible data hiding in encrypted images. IEEE Signal Process. Lett., 2011. 18(4), pp. 255-258.
9. W. Hong , T. Chen, and H. Wu, An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett., 2012. 19(4), pp. 199-202.
10. X. Zhang, Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensics Security, 2012. 7(2), pp. 826-832.
11. J.B. Feng, I.C. Lin, C.S. Tsai, and Y.P. Chu, Reversible watermarking: Current status and key issues. Int. J. Netw. Security, 2006. 12(3), pp. 161-171.
12. K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans. Inf. Forensics Security, 2013. 8(3), pp. 553-562.
13. C.W. Honsinger, P.W. Jones, M. Rabbani, and J.C. Stoffel, Lossless recovery of an original image containing embedded data. U.S. Patent 6,278,791, 2001.
14. T. Kalker and F.M. Willems, Capacity bounds and code constructions for reversible data-hiding. Proc. 14th Int. Conf. Digital Signal Process., 2002. 1, pp. 71-76.
15. X. Chen, X. Sun, H. Sun, Z. Zhou, J. Zhang, Reversible watermarking method based on asymmetric-histogram shifting of prediction errors. J. Syst. Softw., 2013. 86(10), pp. 2620-2626.
16. O.M. Al-Qershi, B.E. Khoo, High capacity data hiding schemes for medical images based on difference expansion. J. Syst. Softw., 2011. 84(1), pp. 105-112.
17. A. Khan, A. Siddiqa, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques. Information Sciences, 2014. 279, pp. 251-272.
18. USC-SIPI Image database [Online]. Available: http://sipi.usc.edu/database/